

LATERAL BUCKLING OF BEAMS WITHOUT
WARPING RIGIDITY

Robert Edward Brown

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

LATERAL BUCKLING OF BEAMS WITHOUT
WARPING RIGIDITY

by

Robert Edward Brown

June 1978

Thesis Advisor:

J. E. Brock

Approved for public release; distribution unlimited.

T184052

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Lateral Buckling of Beams Without Warping Rigidity		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; June 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert Edward Brown		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1978
		13. NUMBER OF PAGES 58
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Lateral buckling; lateral instability, buckling of beams.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A procedure is described to determine lateral buckling loads for initially straight beams. Loading and beam geometry may vary along the length. Warping rigidity is not considered. End conditions of considerable generality may be treated. The algorithm depends on solving a convergent sequence of sixteenth order eigenproblems. A computer program implementing this procedure has been developed and is		

described herein.

Approved for public release; distribution unlimited.

Lateral Buckling of Beams Without
Warping Rigidity

by

Robert Edward Brown
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1968

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the
NAVAL POSTGRADUATE SCHOOL
June 1978

ABSTRACT

A procedure is described to determine lateral buckling loads for initially straight beams. Loading and beam geometry may vary along the length. Warping rigidity is not considered. End conditions of considerable generality may be treated. The algorithm depends on solving a convergent sequence of sixteenth order eigenproblems. A computer program implementing this procedure has been developed and is described herein.

TABLE OF CONTENTS

I.	INTRODUCTION - - - - -	7
II.	PROBLEM SPECIFICATIONS - - - - -	9
III.	STRUCTURAL ANALYSIS - - - - -	17
IV.	MATHEMATICAL ANALYSIS - - - - -	25
V.	COMPUTER IMPLEMENTATION OF THEORY - - - - -	33
VI.	TESTING AND VERIFICATION - - - - -	36
VII.	CONCLUSIONS AND RECOMMENDATIONS - - - - -	40
	APPENDIX A - - - - -	41
	BIBLIOGRAPHY - - - - -	57
	INITIAL DISTRIBUTION LIST - - - - -	58

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Dr. John E. Brock, Professor of Mechanical Engineering, for his invaluable guidance and advice. Without his inspiration and patient assistance this work could not have been accomplished.

II. PROBLEM SPECIFICATIONS

The coordinate system used is a right-handed orthogonal x, y, z system with positive senses as indicated in figure 1. The longitudinal centroidal axis of the beam lies along the $+z$ direction. The principal axes of a normal cross section are originally in the x and y directions but after torsional displacement (rotation) they are in the ξ and η directions, cf. figure 2 (page 10).

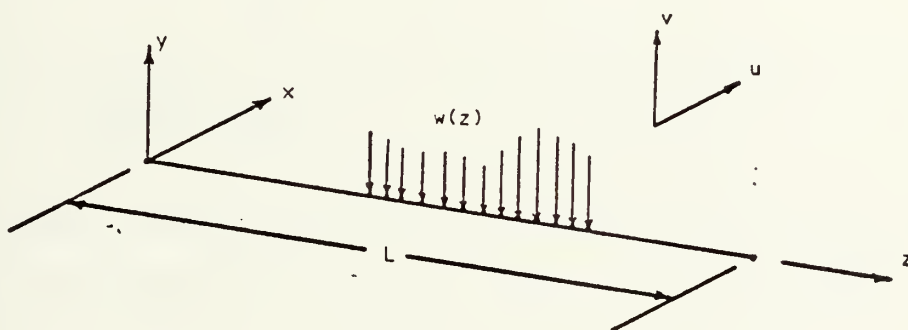


Figure 1. General configuration and coordinate system.

The beam displacements are defined by three variables: vertical deflection v , horizontal deflection u , and angular rotation ϕ . Positive senses for v and u are as indicated in figure 1. The positive sense for ϕ is indicated in figure 2. The deformations u , v and ϕ are considered to be "small" in a mathematical sense, so that, for example $\sin \phi \approx \phi$.

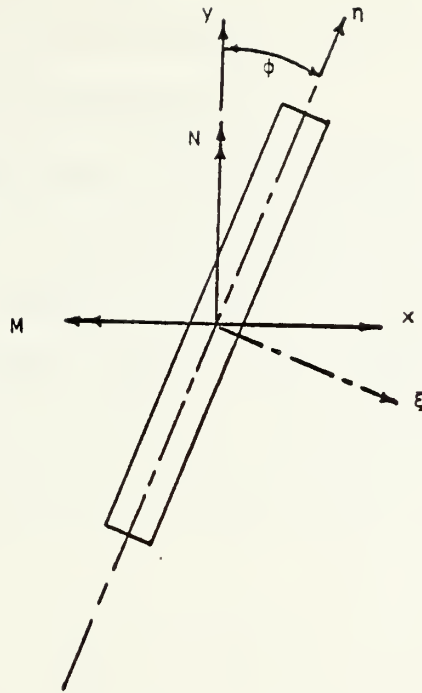


Figure 2. Positive sense of ϕ and source of bending moment $N-\phi M$.

Defining the problem requires specifications for five physical characteristics of the beam. They are beam length L , flexural rigidity for bending in the vertical plane EI_1 ,^{*} flexural rigidity for bending in the horizontal plane EI_2 , torsional rigidity C , and warping rigidity C_1 . The last four may vary along the length of the beam.

Torsional rigidity C is defined by the following equation.

$$C = (\text{TORQUE})/(\text{ANGLE OF TWIST PER UNIT LENGTH}) \quad (1)$$

^{*}For simplicity in notation, we use numerical suffixes rather than numerical subscripts.

Warping rigidity C_1 is related to nonuniform torsion. A discussion here of this property would be too digressive; cf. section 5.3 of reference (2).

It is the underlying purpose of this entire analysis to determine the total externally applied load P which will cause incipient lateral buckling. The only loading considered is distributed load in the negative y direction. The type of loading (uniform, triangular or whatever) is specified beforehand by a dimensionless function $f(z)$ such that

$$w(z) = (R/L^3) f(z) \quad (2)$$

where the load shape function $w(z)$ has dimensions of force per unit length. Concentrated loads and concentrated moments can be approximated by appropriate large local variations in $f(z)$. The function $f(z)$ will be denoted by x_5 when it appears later. The constant R has dimensions of force times distance squared. The magnitude of R is arbitrary and is chosen at the convenience of the user.

The load function has been defined. We are interested in determining the multiplying factor Q such that lateral buckling does not take place if the actual loading is less than $Qw(z)$ and does take place if the actual loading exceeds $Qw(z)$. The entire purpose of the program developed from this analysis is to determine this multiplier Q . In order to obtain nondimensionalized equations, it is convenient to

introduce P the total applied load for incipient buckling, defined by

$$P = Q \int_0^L w(z) dz \quad (3)$$

We will also employ the normalization

$$\int_0^L f(z) dz = L \quad (4)$$

so that $Q = PL^2/R \quad (5)$

For the special case where the load contributions in the negative y direction are canceled by the load contributions in the positive y direction the following alternate form of equation 3 is applied.

$$P = \int_0^L Q |w(z)| dz \quad (6)$$

The load may be applied above, on, or below the centroidal axis as specified by the load eccentricity, $e(z)$, cf. figure 6 (page 20).

For notational and programming convenience a series of dimensionless functions x are defined. The first six are used to specify beam properties and loading. A seventh is simply a function having constant unit value.

$$x_1 = EI_1/R \quad (7)$$

$$x_2 = EI_2/R \quad (8)$$

$$x_3 = C/R \quad (9)$$

$$x_4 = RL^2/C_1 \quad (10)$$

$$x_5 = w(z) L^3/R = f(z) \quad (11)$$

$$x_6 = e(z)/L \quad (12)$$

$$x_7 = 1 \quad (13)$$

The nondimensionalizing scheme used to define the x functions is applied throughout the problem development. The dimensionless forms for vertical and horizontal deflection are

$$x_{10} = v/QL \quad (14)$$

$$\text{and} \quad \beta = u/L \quad (15)$$

where β is a dimensionless function which ultimately will be constructed from several functions.

We will use the dimensionless parameter

$$\zeta = z/L \quad (16)$$

to specify axial position. Accordingly we will think of the x functions as functions of ζ , thus $x_5 = x_5(\zeta)$. Differentiation with respect to z and ζ are indicated as $(\frac{dY}{dz} = Y')$ and $(\frac{dY}{d\zeta} = \dot{Y})$ respectively. The two operations are related by the equation

$$\frac{d}{dz} = \frac{1}{L} \frac{d}{d\zeta} \quad (17)$$

The developments which follow require considerable integration. Two notational devices are used to facilitate both the readability and the transcription of equations. Integration is signified by p, thus

$$px3 = \int_0^{\zeta} x3(\zeta)d\zeta .$$

Evaluation of any function at the right end of the beam, i.e., for $\zeta = 1$, is indicated by the prefix *. Several examples illustrating this notation are

$$*px1 = \int_0^1 x1(\zeta)d\zeta \quad (18)$$

$$*x5x9 = x5(1) \cdot x9(1) \quad (19)$$

$$*x2 \ px8 = x2(1) \int_0^1 x8(\zeta)d\zeta \quad (20)$$

$$*px5x6px11 = \int_0^1 x5(\zeta)x6(\zeta) \left[\int_0^{\zeta} x11(\theta)d\theta \right] d\zeta \quad (21)$$

(In the last equation θ is a dummy variable.)

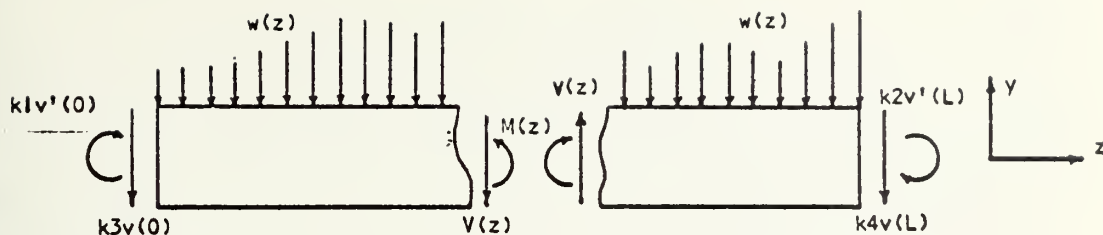


Figure 3. Diagram of forces and moments in the vertical plane.

Our analysis is restricted to beams which are unconstrained except at one or both ends. There are six possible constraints at each end each of which is modeled as a spring. The spring constant k describing such a constraint can actually vary from zero (perfectly free, no constraint at all) to infinity (perfect or ideal constraint) inclusive. Therefore to avoid mathematical difficulties the analysis employs dimensionless spring parameters, designated by the symbol a (with an appropriate identifying suffix). For a torsion spring, such as that corresponding to the moment $k_1 v'(0)$ shown in figure 3 (page 15), the relations between k_1 and a_1 are

$$a_1 = k_1 / (k_1 + R/L) \quad (22)$$

$$k_1 = R a_1 / L (1 - a_1) \quad (23)$$

Note that $a_1 = 0$ corresponds to $k_1 = 0$ (perfect freedom) and $a_1 = 1$ corresponds to $k_1 = \infty$ (perfect constraint).

Similarly, for a linear spring such as spring number 3, the relations are

$$a_3 = k_3 / (k_3 + R/L^3) \quad (24)$$

$$k_3 = Ra_3/L^3(1 - a_3) \quad (25)$$

Figure 5 (page 19) shows that the point of application of a linear spring is not limited to the centroidal axis. The vertical deflection constraints may be applied at distances c_1 and c_2 above the centroidal axis, and the horizontal deflection constraints may be applied at distances c_3 and c_4 above the centroidal axis. The positive sense is as indicated in figure 5, and the dimensionless forms of these distances are

$$G_1 = c_1/L \quad (26)$$

$$G_2 = c_2/L \quad (27)$$

$$G_3 = c_3/L \quad (28)$$

$$G_4 = c_4/L \quad (29)$$

The physical and geometrical parameters used in the problem have now been defined. They will be used in the structural analysis developed in the next section.

III. STRUCTURAL ANALYSIS

There are three structural deformations examined in the analysis. They are vertical bending, horizontal bending, and torsion.

The vertical bending problem is disjoint from the remainder of the analysis, and it is solved first. The notation used in defining vertical bending will not be shown because the problem is simple and familiar. It is sufficient to say that a set of four simultaneous nonhomogeneous equations are formed, representing the constraint conditions at each end which pertain to bending in the vertical (yz) plane. This system is then solved by a standard algebraic analysis, and the result is used to create three dimensionless functions for shear, moment, and deflection.

$$\text{Shear} \quad x8 = V/P \quad (30)$$

$$\text{Moment} \quad x9 = M/PL \quad (31)$$

$$\text{Deflection} \quad x10 = v/QL \quad (32)$$

The remainder of the analysis involves studying the combined horizontal bending and torsion problem. Figure 4 (page 18) shows the positive sense for the constant horizontal shear H , the linearly varying horizontal bending moment N , and horizontal displacement u .

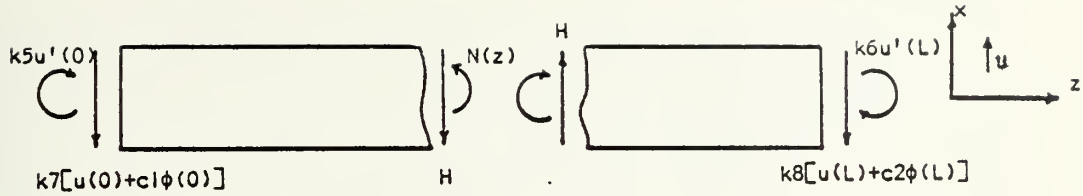


Figure 4. Diagram showing forces and moments in the horizontal plane.

The appropriate dimensionless variables are

$$\beta = u/L ; \quad u = L\beta \quad (33)$$

$$B4 = H/P ; \quad H = PB4 = B4P \quad (34)$$

$$n = N/PL; \quad N = PLn \quad (35)$$

Dimensionless shear $B4$ is one of ultimately eight unknown scalar quantities ($B1, B2, \dots, B8$) which will be introduced in the process of determining Q . The alternate forms indicated in the preceding equations are intended to familiarize the reader with notational forms which will be freely employed in later developments.

The illustration of the rotated cross section in figure 2 (page 10) shows that the moment causing bending about the η axis is $N \cos \phi - M \sin \phi$, which reduces to $N - M\phi$ by small angle approximation. From elementary bending theory we have

$$EI2 \, u'' = N - \phi M \quad (36)$$

and by substitution from equation 7 it may be written as

$$\ddot{\beta} = Qx^2(n - \phi x^9) \quad (37)$$

Examination of the simple statics of figure 4 (page 18) yields

$$Hz + N(0) = N(z) \quad (38)$$

In dimensionless form this becomes

$$B_4\zeta + n(0) = n(\zeta) \quad (39)$$

and may be rewritten as

$$n = B_4px^7 + B_5x^7 \quad (40)$$

$$n(0) = B_5 \quad (41)$$

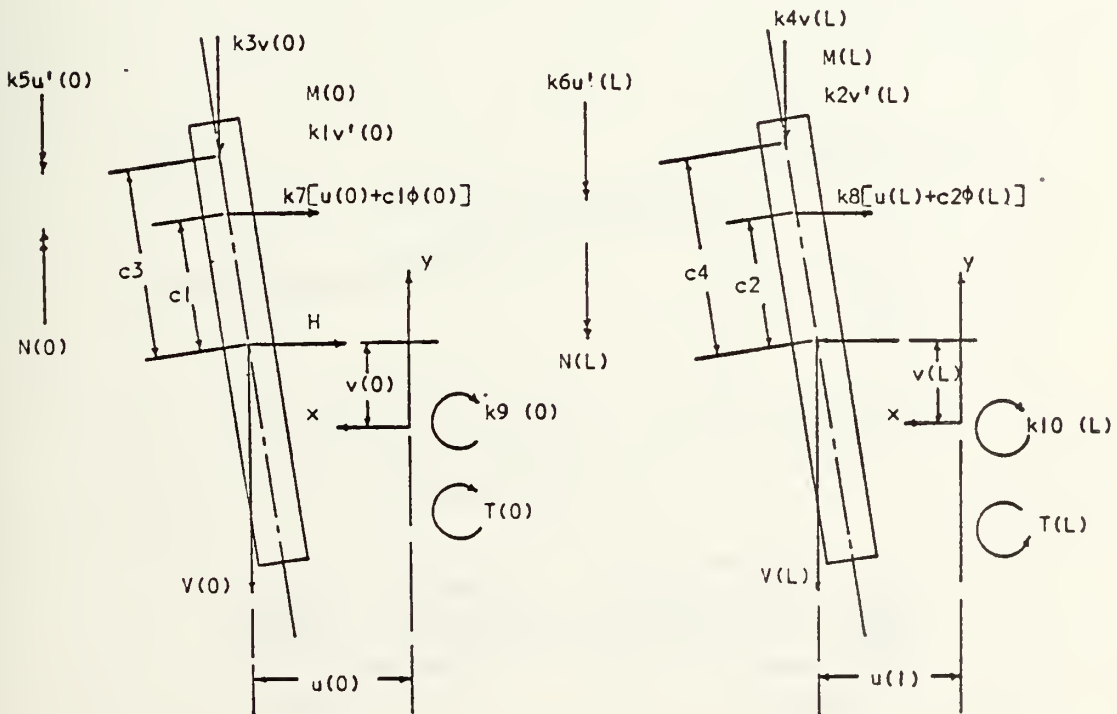


Figure 5. Force and moment diagrams of end sections.

Figures 5 and 6 (page 20) show the forces and moments used to develop the torsion moment equation

$$T(z) = T(0) - H[v(z) - v(0)] + V(z)u(z) - V(0)u(0) + \int_0^z w(z)[u(z) + e(z)\phi(z)]dz \quad (42)$$

Introducing the dimensionless function

$$t = T/PL \quad (43)$$

and using additional dimensionless functions previously defined, equation 31 may be written as

$$t = t(0) - QB4[x10 - x10(0)] + x8\beta - x8(0)\beta(0) + px5(\beta + x6\phi) \quad (44)$$

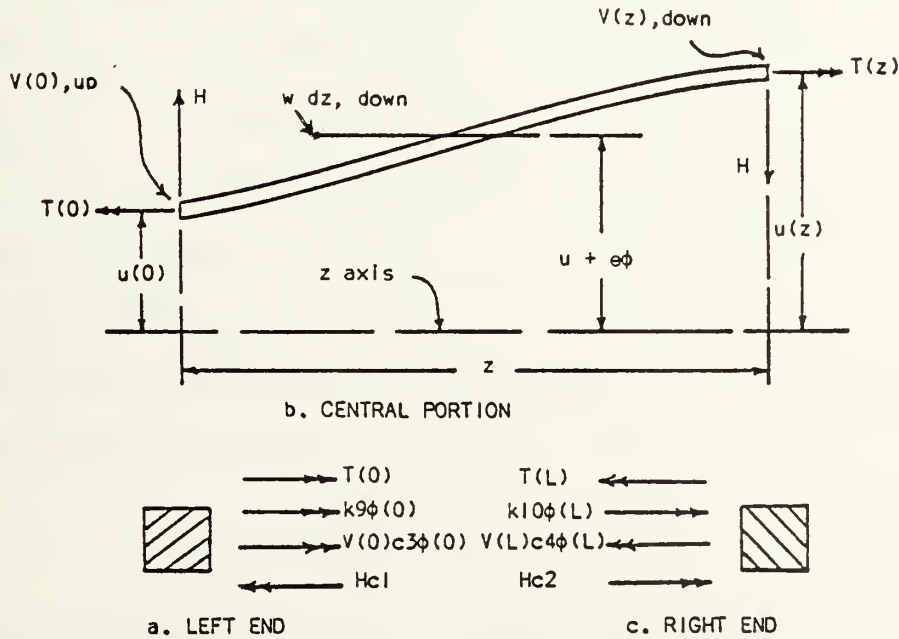


Figure 6. Diagram of left, central, and right portions for torsion analysis.

Reference (1) develops the following third order equation which accounts for the warping rigidity of the beam.

$$C_1 \phi''' = C \phi' + T - u'M \quad (45)$$

In dimensionless form this may be written as

$$\ddot{\phi} = x_4(Q_t - Qx_9\dot{\beta} + x_3\dot{\phi}) \quad (46)$$

The function x_4 has previously been defined as

$$x_4 = RL^2/C_1 \quad (47)$$

C_1 is the coefficient of the highest (i.e., third) order derivative in equation 45, and if C_1 becomes small, very troublesome mathematical difficulties are encountered, cf. Reference (3). Accordingly, the analysis for the case $C_1 \equiv 0$ cannot be obtained from that for $C_1 \neq 0$, given in reference (1), except with the greatest of difficulty. It is for this reason that the separate analysis represented by this thesis, was undertaken. The analysis herein, to this point, is, as has been said before, essentially identical to that in reference (1). From this point on there are essential differences.

The present analysis presumes that $C_1 = 0$, so that equation 35 takes the form

$$\dot{\phi} = Qx_3^{-1}(x_9\dot{\beta} - t) \quad (48)$$

The inverse of x_3 , x_3^{-1} , occurs frequently in what follows. The term is clumsy in this form and will be redesignated as

$$Y = x_3^{-1} \quad (49)$$

Therefore equation 48 is written as

$$\dot{\phi} = QY(x_9\dot{\beta} - t) \quad (50)$$

Two of the end constraint conditions of the original problem, namely those relating to warping constraints at the end, are not available for and do not appear in the present specialized problem.

The original problem had six constraints at each end. The present problem has five constraints at each end. Four constraint conditions were employed in solving the problem of bending in the vertical plane. Thus a total of six constraint conditions remain at this stage of the development. The first four result from the constraints against horizontal deflection and end rotations in the x-z plane. The last two result from constraints against rotation about the z axis.

$$a_5\dot{\beta}(0) - Q(1-a_5) n(0) = 0 \quad (51)$$

$$a_6\dot{\beta}(1) + Q(1-a_6) n(1) = 0 \quad (52)$$

$$a_7[\beta(0)+G_1\phi(0)] + QB_4(1-a_7) = 0 \quad (53)$$

$$a_8[\beta(1)+G_2\phi(1)] - QB_4(1-a_8) = 0 \quad (54)$$

$$a_9\phi(0)+Q(1-a_9)[t(0)+G_3x_8(0)\phi(0)-B_4G_1] = 0 \quad (55)$$

$$a_{10}\phi(1)-Q(1-a_{10})[t(1)+G_4x_8(1)\phi(1)-B_4G_2] = 0 \quad (56)$$

Although this notation appears formidable, it is convenient, explicit, and unequivocal. The terms are

sufficiently complicated that no matter what notation might be used careful attention to detail is required.

There is no need to demonstrate the complete development of the six boundary equations, but for illustrative purposes the derivations of two typical constraint equations, 51 and 53, will be given.

The constraint against rotation of the left end about a vertical (y) axis, as shown in figure 6 (page 20) results in the following equation.

$$k_5 u'(0) - N(0) = 0 \quad (57)$$

Employing the dimensionless variable $\beta = u/L$ and $n = N/PL$, using equation 17, writing the spring constant k_5 in terms of the spring parameter a_5 , viz,

$$k_5 = Ra_5/L(1-a_5) \quad (58)$$

and recalling equation 5, it is easy to obtain the equation

$$a_5 \dot{\beta}(0) - Q(1-a_5)n(0) = 0 \quad (59)$$

This is equation 51.

Equation 53 defines the constraint against horizontal deflection of the left end of the beam. The deflection is $u(0) + c_1 \phi(0)$, see figure 5 (page 19), where the second term results from permitting the spring to be attached at a distance c_1 above the centroidal axis. The constraint equation is

$$k_7 [u(0) + c_1 \phi(0)] + H = 0 \quad (60)$$

where as previously defined $H = PB_4$, $c_1 = LG_1$, and

$$k_7 = a_7 R / L^3 (1 - a_7) \quad (61)$$

Using the dimensionless quantities previously introduced, it is a simple matter to put equation 60 in the form exhibited as equation 53.

The structural analysis has now been completed and is represented by equations 37, 44, 50, 51, 52, 53, 54, 55, and 56. The mathematical method of solution will be addressed next.

IV. MATHEMATICAL ANALYSIS

Before going into the details of the analysis, we will try to describe it in general terms.

An expression is formed for $\dot{\phi}$ which involves an unknown function (x_{11} in what follows) the initial form of which is arbitrarily assumed except for the normalizations

$$x_{11}(0) = 0 \quad (62)$$

$$*p|x_{11}| = 1 \quad (63)$$

The corresponding function ϕ is obtained by integration and equation 37 is used to get $\ddot{\beta}$. This is followed by two more integrations to get $\dot{\beta}$ and β . These are employed in the torsion equation 44 to get the function t . Finally this is used in equation 48 to get a new expression for $\dot{\phi}$. Each integration introduces a new unknown scalar quantity. In one way or another a total of eight such unknown scalar constants B_1, B_2, \dots, B_8 are introduced. The six constraint equations, plus two more which will be introduced in what follows, provide a set of eight linear homogeneous equations in these unknown B's which thus leads to an eigenproblem of eighth order. The parameter Q appears in the coefficients to the first and second powers. The quadratic eighth order eigenproblem is transformed to a linear sixteenth order eigenproblem. Since, generally, both zero and infinite eigenvalues are contained in the solution, a special, relatively new

algorithm, the QZ method, capable of dealing with such systems, is employed for the solution of this eigenproblem. The appropriate eigenvalue is chosen and the corresponding eigenvector provides the solution for the unknown B's so that the new x_{11} can be uniquely determined. This new x_{11} is used in place of the original x_{11} and the process is repeated until there is convergence. No proof is offered that indeed convergence must ultimately take place; because of the complexity of the problem such a proof might be very difficult to establish. However, experience with the procedure described here indicates that convergence may indeed be expected. The details of this procedure are now given in what follows.

The function $\dot{\phi}$ is initially specified as

$$\dot{\phi} = B_1 x_{11} + B_2 x_7 \quad (64)$$

where x_{11} is an arbitrarily assumed function satisfying equations 62 and 63 and where B_1 and B_2 are the third and fourth of eight unknown constants that will be employed. The first was B_4 from equation 34 and the second was B_5 from equation 41.

Integrating equation 64 yields

$$\phi = B_1 p x_{11} + B_2 p x_7 + B_3 x_7 \quad (65)$$

and when equations 40 and 65 are substituted into equation 37 we arrive at

$$\ddot{\beta} = -B1Qx2x9px11 - B2Qx2x9px7 - B3Qx2x9px7 + B4Qx2px7 + B5Qx2 \quad (66)$$

Integrating twice more, we get

$$\dot{\beta} = -B1Qpx2x9px11 - B2Qpx2x9px7 - B3Qpx2x9px7 + B4Qpx2px7 + B5Qpx2 + B6Qx7 \quad (67)$$

$$\beta = -B1Qppx2x9px11 - B2Qppx2x9px7 - B3Qppx2x9px7 + B4Qppx2px7 + B5Qppx2 + B6Qpx7 + B7Qx7 \quad (68)$$

For convenience the next step is to rewrite equation 44 as

$$t = q\beta + px5x6\phi - QB4X - Qx8(0)B7 + QB8 \quad (69)$$

$$\text{where} \quad t(0) = QB8 \quad (70)$$

$$X = x10 - x10(0) \quad (71)$$

$$q = px5 + x8 \quad (72)$$

The symbol q denotes an operator acting on the function β , not a function which is multiplied by β .

When the preceding equations, evaluated at $\zeta=0$ or $\zeta=1$ as the case may be, are substituted in boundary conditions equations 51 through 56, a system of six linear simultaneous homogeneous equations in eight unknown constants B is obtained.

In the previous section, which addressed the structural analysis, equations 51 and 53 were derived for illustrative purposes. Now for illustrative continuity these two equations will be used to show the method of formulation of the six linear simultaneous homogeneous equations.

Recalling that equation 51 is

$$a_5 \dot{\beta}(0) - Q(1-a_5) n(0) = 0$$

and substituting

$$\dot{\beta}(0) = QB_6 \quad (73)$$

$$n(0) = B_5 \quad (74)$$

we now factor out Q leaving

$$-B_5(1-a_5) + B_6a_5 = 0 \quad (75)$$

The second illustrative equation, 53, is

$$a_7[\beta(0) + G_1\phi(0)] + QB_4(1-a_7) = 0$$

and upon substituting

$$\phi(0) = B_3 \quad (76)$$

$$\beta(0) = B_7Q \quad (77)$$

we arrive at

$$B_3G_1a_7 + B_4Q(1-a_7) + B_7Qa_7 = 0 \quad (78)$$

If equations 67 and 69 are substituted into equation 50 the following "new" $\dot{\phi}$ is created.

$$\begin{aligned}
\dot{\phi} = & Y[B1(-Q^2x9px2x9px11 + Q^2qppx2x9px11 - Qpx5x6px11) \\
& + B2(-Q^2x9px2x9px7 + Q^2qppx2x9px7 - Qpx5x6px7) \\
& + B3(-Q^2x9px2x9 + Q^2qppx2x9 - Qpx5x6) \\
& + B4(Q^2x9px2px7 + Q^2X - Q^2qppx2px7) \\
& + B5(Q^2x9px2 - Q^2qppx2) \\
& + B6(Q^2x9 - Q^2qpx7) \\
& + B8(-Q^2)]
\end{aligned} \tag{79}$$

(The coefficient of B7 vanishes identically.)

To complete the eigensystem involving the eight unknown B's, two additional equations are needed.

The seventh equation, consistent with equation 62, is

$$\text{new } \dot{\phi}(0) - B2 = 0 \tag{80}$$

The eighth equation is

$$(*p\dot{\phi})_{\text{old}} - (*p\dot{\phi})_{\text{new}} = 0 \tag{81}$$

and is arrived at by observing that if $\dot{\phi}$ given in equation 64 and created from an assumed x11, were, somehow, a "correct" $\dot{\phi}$ then it and the $\dot{\phi}$ recovered in equation 79 would be identically equal to each other. The integration in equation 81 achieves a certain "smoothing" or averaging which has been found to be preferable to simply equating the new and old functions at an arbitrary value of ζ .

The coefficients of the eigensystem equations generally involve Q to the zeroth, first, and second powers so that the system may be exhibited in the form

$$(D + EQ + FQ^2)B = 0 \tag{82}$$

The nonzero elements of matrices D, E, and F are

$D_{15} = a_5 - 1$, $D_{16} = a_5$, $D_{21} = -a_6 * p_{x2} * x_9 * p_{x11}$,
 $D_{22} = -a_6 * p_{x2} * x_9 * p_{x7}$, $D_{23} = -a_6 * p_{x2} * x_9$, $D_{24} = a_6 * p_{x2} * p_{x7} - a_6 + 1$,
 $D_{25} = a_6 * p_{x2} - a_6 + 1$, $D_{26} = a_6$, $D_{33} = G_{1a7}$,
 $D_{41} = a_8 G_2 * p_{x11}$, $D_{42} = a_8 G_2$, $D_{43} = a_8 G_2$,
 $D_{53} = a_9$, $D_{61} = a_{10} * p_{x11}$, $D_{62} = a_{10}$, $D_{63} = a_{10}$,
 $D_{72} = -1$, $D_{81} = - * p_{x11}$, $D_{82} = -1$, $E_{34} = 1 - a_7$,
 $E_{37} = a_7$, $E_{41} = -a_8 * p_{x2} * x_9 * p_{x11}$,
 $E_{42} = -a_8 * p_{x2} * x_9 * p_{x7}$, $E_{43} = -a_8 * p_{x2} * x_9$,
 $E_{44} = a_8 * p_{x2} * p_{x7} + a_8 - 1$, $E_{45} = a_8 * p_{x2}$,
 $E_{46} = a_8$, $E_{47} = a_8$, $E_{53} = (1 - a_9) G_{3x8}(0)$,
 $E_{54} = (a_9 - 1) G_1$, $E_{61} = (a_{10} - 1) [* p_{x5} * x_6 * p_{x11} + G_4 * x_8(1) * p_{x11}]$,
 $E_{62} = (a_{10} - 1) [* p_{x5} * x_6 * p_{x7} + G_4 * x_8(1)]$,
 $E_{63} = (a_{10} - 1) [* p_{x5} * x_6 + G_4 * x_8(1)]$, $E_{64} = (1 - a_{10}) G_2$,
 $E_{81} = - * p_Y * p_{x5} * x_6 * p_{x11}$, $E_{82} = - * p_Y * p_{x5} * x_6 * p_{x7}$,
 $E_{83} = - * p_Y * p_{x5} * x_6$, $F_{58} = 1 - a_9$,
 $F_{61} = (1 - a_{10}) * q_{ppx2} * x_9 * p_{x11}$, $F_{62} = (1 - a_{10}) * q_{ppx2} * x_9 * p_{x7}$,
 $F_{63} = (1 - a_{10}) * q_{ppx2} * x_9$, $F_{64} = (1 - a_{10}) (*X - *q_{ppx2} * p_{x7})$,
 $F_{65} = (a_{10} - 1) * q_{ppx2}$, $F_{66} = (a_{10} - 1) * q_{px7}$,
 $F_{68} = a_{10} - 1$, $F_{76} = x_9(0) Y(0)$, $F_{78} = -Y(0)$,
 $F_{81} = - * p_Y * x_9 * p_{x2} * x_9 * p_{x11} + * p_Y q_{ppx2} * x_9 * p_{x11}$,
 $F_{82} = - * p_Y * x_9 * p_{x2} * x_9 * p_{x7} + * p_Y q_{ppx2} * x_9 * p_{x7}$,
 $F_{83} = - * p_Y * x_9 * p_{x2} * x_9 + * p_Y q_{ppx2} * x_9$,
 $F_{84} = * p_Y * x_9 * p_{x2} * p_{x7} + * p_Y X - * p_Y q_{ppx2} * p_{x7}$,
 $F_{85} = * p_Y * x_9 * p_{x2} - * p_Y q_{ppx2}$,
 $F_{86} = * p_Y * x_9 - * p_Y q_{px7}$,
 $F_{88} = - * p_Y$

[For reference purposes all the preceding equations bear the number 84.]

This eighth order quadratic eigensystem may be expanded to a sixteenth order linear eigensystem in the form

$$VZ = QWZ \quad (85)$$

where

$$V = \begin{bmatrix} E & D \\ I & O \end{bmatrix} \quad (86)$$

$$W = \begin{bmatrix} -F & O \\ O & I \end{bmatrix} \quad (87)$$

$$Z = \begin{bmatrix} QB \\ B \end{bmatrix} \quad (88)$$

and I and O respectively represent the eighth order unit and null matrices.

The eigenproblem has now been defined in a linear form, but because the matrices V and W may be singular not just any method will satisfactorily solve the problem. We have chosen the QZ method described in reference (3). From the sixteen eigenvalues produced the smallest positive eigenvalue is selected. The last eight elements of the associated eigenvector are the desired constants B1 through B8.

Using equation 79 a new $\dot{\phi}$ is now formed and from it a new x11 is created by means of the following normalization process.

$$x11_{new} = [\dot{\phi}_{new} - \dot{\phi}_{new}^{(0)}] / [*P|\dot{\phi}_{new} - \dot{\phi}_{new}^{(0)}|] \quad (89)$$

The iteration process is now repeated until satisfactory convergence is attained. The converged smallest positive eigenvalue Q is now used to determine the buckling load from either equation 3 or equation 6.

V. COMPUTER IMPLEMENTATION OF THEORY

To this point the functions x have been treated as continuous functions for use in an analytic solution. However, it is obvious that solving the problem, even for simple functions, by analytic means is prohibitively difficult. To overcome this problem a numerical method of solution was employed. The beam was divided into a finite number of equal sections. The functions x were then redefined as vectors with Fortran designations $x(1,I)$, $x(2,I)$..., etc. where $I = 1, 2, 3, \dots, N$. The elements of any such vector represent function values at the N points of subdivision of the beam into equal subsections of length $L/(N-1)$. The vector $x(10,I)$ was used twice, first to represent x_{10} and then to store X of equation 71.

The vectors in this form readily lend themselves to manipulation. The simple subroutines developed to perform the operations are, with the exception of the integration scheme, not worthy of note here.

Trapezoidal integration was first used during the debugging of the raw, untested program. This method gives satisfactory results with an error of $O(h^2)$, where h is the interval length. Once it was established that the program would give satisfactory solutions the refinement process was begun. A higher order numerical integration scheme was employed next. It is described by Milne in reference (5),

and results in an error $O(h^5)$.

The QZ subroutine is a canned program in the library of the Naval Postgraduate School Computer Center. It was transcribed from reference (6), with very little change, by Mr. Roger Hilleary. To date, it has been the only program that has given satisfactory solutions to the present eigenvalue problem.

The complete theory described in the preceding sections of this thesis has been implemented in the form of a subroutine LATBRO, a complete listing of which is contained in Appendix A.

The user must write a main program which supplies dimensioning statements, number of intervals, specifications for vectors x_1 through x_6 , specifications for the constants G , and the ten spring parameters. The following is an example of such a main program. It was used to solve what is listed as Case 6, Table 1, page 37.

```
IMPLICIT REAL*8(A-H,O-Z)
REAL*8 X(30,101),ALFA(12),G(4),S,ONE,ZERO
INTEGER KP(10)
COMMON X,ALFA,G,N,KP
N=101
ZERO=0.D+0
ONE=1.D+0
DO 1 I=1,N
X(1,I)=ONE
X(2,I)=ONE
X(3,I)=ONE
X(5,I)=ZERO
X(6,I)=ZERO
1 CONTINUE
X(5,1)=ONE
DO 2 I=1,4
2 G(I)=ZERO
DO 3 I=1,12
```



```
3 ALFA(1)=ZERO
  ALFA(2)=ONE
  ALFA(4)=ONE
  ALFA(6)=ONE
  ALFA(8)=ONE
  ALFA(10)=ONE
  KP(3)=1
  CALL LATBRO(S)
  STOP
  END
```


VI. TESTING AND VERIFICATION

As mentioned in the previous section, the numerical integration methods that have been employed have their shortcomings.

If the integrand is "smooth," either the trapezoidal or the higher order Milne method may be used. The latter is preferable because of its greater accuracy.

Both methods introduce errors, however, when there are steps, impulses, or doublets to be integrated. We have found that the Milne and trapezoidal methods can be used to solve problems with a concentrated load. The error depends linearly on the reciprocal of the number of elements into which the beam is divided and extrapolation may be used to obtain an excellent result. As yet we do not have experience with concentrated moment loadings (doublets). This subject is discussed in more detail in reference (1).

The following table shows the cases that have been tested to date. In each of these cases all the G's and the eccentricity x_6 were zero. Each of these has been successful.

The values for Q shown in the last column of the table were obtained with $x_2 \equiv 1$, $x_3 \equiv 1$, and by simple scaling analysis lead to total load

$$P = Q \sqrt{CEI^2/L^2} \quad (89)$$

TABLE I

Cases of Uniform Beams Analyzed by Subroutine LATBRO
($x6 = 0$, $G1 = G2 = G3 = G4 = 0$)

Case Number	Values of Spring Parameters										Loading Type	Q
	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10		
1	0	1	0	1	0	1	0	1	0	1	UNIF	12.854
2	1	0	1	0	1	0	1	0	1	0	UNIF	12.854
3	0	0	1	1	1	1	1	1	1	1	UNIF	47.58
4	1	1	1	1	0	0	1	1	1	1	UNIF	97.11
5	0	0	1	1	0	0	1	1	1	1	CONC MIDSPAN	16.936
6	0	1	0	1	0	1	0	1	0	1	CONC LEFT END	4.018

for lateral buckling. The values reported agree with values to be found in reference (2).

The following are details of several cases.

Case 1

As shown, it is a cantilever beam fully restrained at the right end. The loading is uniform.

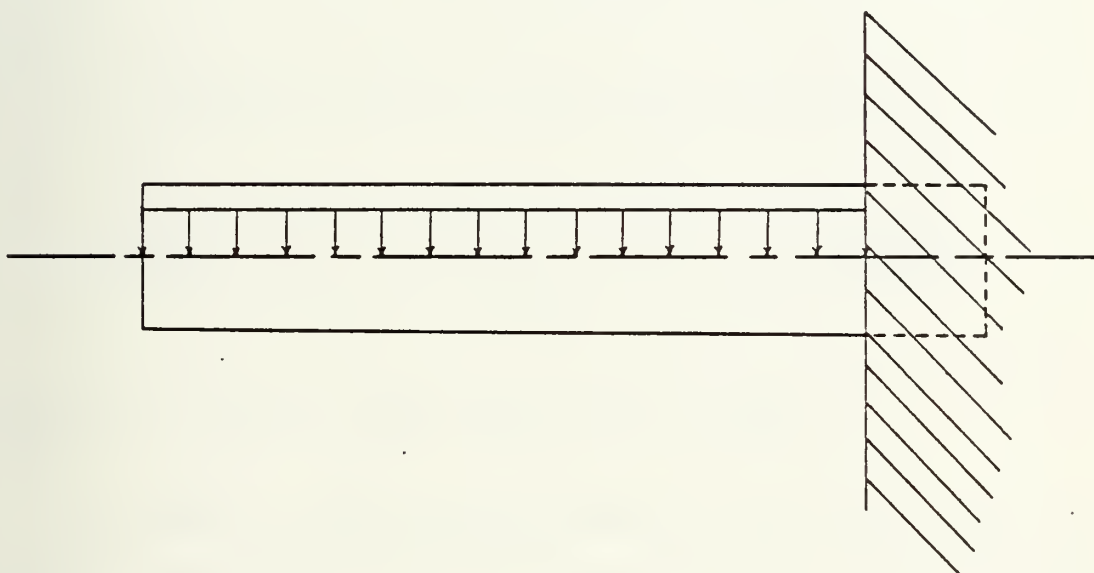


Figure 7. Illustration of side view of Case 1 beam.

The subroutine produced a converged value of Q of 12.854.

Timoshenko's result is 12.85; cf. page 261 ref (2).

Case 2

This is simply Case 1 turned end for end. It was used to test additional equations and also resulted in a converged critical load output of 12.854 which is identical to the result for Case 1, as it should be.

Case 5

The converged critical load output was 16.936 and Timoshenko's result is 16.94; cf. page 269 ref (2).

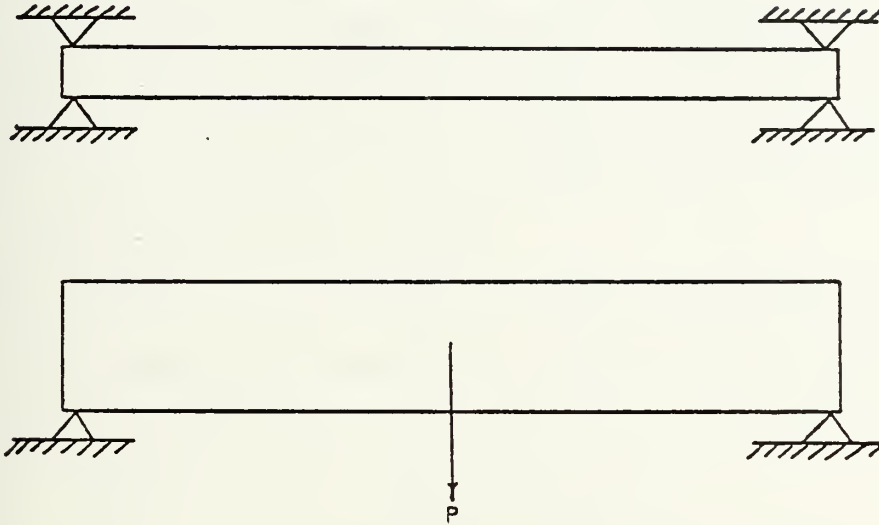


Figure 8. Top and side views Case 5 beam.

In several of the cases listed in the table above, the loading does not interact with one or more of the springs, so that the corresponding spring parameters could have an arbitrary value without changing the results.

VII. CONCLUSIONS AND RECOMMENDATIONS

The subroutine LATBRO in its present form should be able to solve beam problems with beams that have uniform or variable cross sections, with loading on the centroidal axis, and with spring constraints attached at the centroidal axis. Loading that involves concentrated moments, as mentioned before, has not yet been tested.

The program was designed to solve problems where loading and spring constraints may act away from the centroidal axis. It is recommended that follow on testing begin with

1. loading off the centroidal axis.
2. spring constraints attached away from the centroidal axis.
3. concentrated moment loading.

It is expected, with the exception of minor programming errors, that problems with any combination of the above will be readily solved by the subroutine.

Listing of LATBRO and Adjunct Subroutines

41


```

D(1,5)=ALFA(5)-ONE
D(1,6)=ALFA(5)
D(2,6)=ALFA(6)
D(3,3)=ALFA(7)*G(1)
E(3,4)=ONE-ALFA(7)
E(3,7)=ALFA(7)
D(4,3)=ALFA(8)*G(2)
D(4,2)=ALFA(8)*G(2)
E(4,6)=ALFA(8)
E(4,7)=ALFA(8)
E(5,3)=(ONE-ALFA(9))*G(3)*X(8,1)
D(5,3)=ALFA(9)
E(5,4)=(ALFA(9)-ONE)*G(1)
F(5,8)=ONE-ALFA(9)
D(6,2)=ALFA(10)
D(6,3)=ALFA(10)
E(6,4)=(ONE-ALFA(10))*G(2)
F(6,8)=ALFA(10)-ONE
D(7,2)=ONEG
D(8,2)=ONEG
CALL DIVV(7,3,14)
F(7,8)=-X(14,1)
F(7,6)=X(14,1)*X(9,1)
CALL INTV(14,15)
F(8,8)=-X(15,N)
CALL INTV(7,12)
CALL MULV(9,12,13)
CALL MULV(2,13,20)
CALL INTV(20,19)
D(2,2)=-ALFA(6)*X(19,N)
CALL INTV(19,17)
CALL QVEC(17,18)
E(4,2)=-ALFA(8)*X(17,N)
F(6,2)=(ONE-ALFA(10))*X(18,N)
CALL MULV(2,9,14)
D(2,3)=-ALFA(6)*X(15,N)
CALL INTV(15,16)
E(4,3)=-ALFA(8)*X(16,N)
CALL QVEC(16,17)
F(6,3)=(ONE-ALFA(10))*X(17,N)
CALL MULV(2,12,14)
CALL INTV(14,15)
D(2,4)=ALFA(6)*X(15,N)-ALFA(6)+ONE
CALL INTV(15,16)
E(4,4)=ALFA(8)*X(16,N)+ALFA(8)-ONE
CALL INTV(2,14)
D(2,5)=ALFA(6)*X(14,N)-ALFA(6)+ONE

```

```

BRK11720
BRK11730
BRK11740
BRK11750
BRK11760
BRK11770
BRK11780
BRK11790
BRK11800
BRK11810
BRK11820
BRK11830
BRK11840
BRK11850
BRK11860
BRK11870
BRK11880
BRK11890
BRK11900
BRK11910
BRK11920
BRK11930
BRK11940
BRK11950
BRK11960
BRK11970
BRK11980
BRK11990
BRK12000
BRK12010
BRK12020
BRK12030
BRK12040
BRK12050
BRK12060
BRK12070
BRK12080
BRK12090
BRK12100
BRK12110
BRK12120
BRK12130
BRK12140
BRK12150
BRK12160
BRK12170
BRK12180
BRK12190

```



```

C * * * * * CALL INTV(7,11) BRK13160
C * * * * * CONTINUE BRK13170
C * * * * * CALL PRIV(11,2,11) BRK13180
C * * * * * CALL PRIV(11,2,11) BRK13190
C * * * * * CALL INTV(11,12) BRK13200
C * * * * * CALL INTV(9,12,13) BRK13210
C * * * * * CALL MULV(2,13,14) BRK13220
C * * * * * CALL INTV(14,15) BRK13230
C * * * * * D(2,1)=-ALFA(6)*X(15,N) BRK13240
C * * * * * CALL INTV(15,16) BRK13250
C * * * * * E(4,1)=-ALFA(8)*X(16,N) BRK13260
C * * * * * D(4,1)=ALFA(8)*G(2)*X(12,N) BRK13270
C * * * * * D(6,1)=ALFA(10)*X(12,N) BRK13280
C * * * * * CALL QVEC(16,17) BRK13290
C * * * * * F(6,1)=(ONE-ALFA(10))*X(17,N) BRK13300
C * * * * * D(8,1)=-X(12,N) BRK13310
C * * * * * CALL MULV(9,15,18) BRK13320
C * * * * * CALL SUBV(17,18,19) BRK13330
C * * * * * CALL DIVV(19,3,20) BRK13340
C * * * * * CALL INTV(20,13) BRK13350
C * * * * * F(8,1)=X(13,N) BRK13360
C * * * * * CALL MULV(6,12,13) BRK13370
C * * * * * CALL MULV(5,13,14) BRK13380
C * * * * * CALL INTV(14,15) BRK13390
C * * * * * S=X(8,N) BRK13400
C * * * * * CALL MULS(12,16,S) BRK13410
C * * * * * E(6,1)=(ALFA(10)-ONE)*(X(15,N)+(G(4)*X(16,N))) BRK13420
C * * * * * CALL DIVV(15,3,17) BRK13430
C * * * * * CALL INTV(17,18) BRK13440
C * * * * * E(8,1)=-X(18,N) BRK13450
C * * * * * 29 FORMAT(//) BRK13460
C * * * * * WRITE(6,29) BRK13470
C * * * * * DC 500 I=1,8 BRK13480
C * * * * * WRITE(6,29) BRK13490
C * * * * * DO 500 J=1,8 BRK13500
C * * * * * WRITE(6,501) D(I,J),E(I,J),F(I,J) BRK13510
C * * * * * 500 CONTINUE BRK13520
C * * * * * 501 FCRMAT(25X,1P3E20.8) BRK13530
C * * * * * WRITE(6,29) BRK13540
C * * * * * DC 500 I=1,8 BRK13550
C * * * * * WRITE(6,29) BRK13560
C * * * * * DC 500 J=1,8 BRK13570
C * * * * * WRITE(6,501) D(I,J),E(I,J),F(I,J) BRK13580
C * * * * * 500 CONTINUE BRK13590
C * * * * * BRK13600
C * * * * * BRK13610
C * * * * * BRK13620
C * * * * * BRK13630

```


[illegible]


```

C * * * * * CALL INTV(11,12)
C * * * * * CALL MULV(9,12,13)
C * * * * * CALL MULV(6,12,14)
C * * * * * CALL MULV(2,13,15)
C * * * * * CALL MULV(5,14,16)
C * * * * * CALL INTV(15,17)
C * * * * * CALL INTV(17,18)
C * * * * * CALL QVEC(18,19)
C * * * * * CALL INTV(16,20)
C * * * * * CALL MULV(9,17,12)
C * * * * * CALL SUBV(19,12,13)
C * * * * * S=LAM**2
C * * * * * CALL MULS(13,14,S)
C * * * * * S=LAM
C * * * * * CALL MULS(20,15,S)
C * * * * * CALL SUBV(14,15,16)
C * * * * * S=B(1)
C * * * * * CALL MULS(16,17,S)
C * * * * * CALL DUPV(17,12)
C * * * * *
C * * * * * VECTOR X12 NOW CONTAINS THE B1 PORTION OF PHI-DOT. THE PORTIONS
C * * * * * WITH THE OTHER CONSTANTS WILL BE ADDED TO VECTOR X12 AS THEY ARE
C * * * * * DEVELOPED.
C * * * * *
C * * * * * CALL INTV(7,13)
C * * * * * CALL MULV(9,13,14)
C * * * * * CALL MULV(6,13,15)
C * * * * * CALL MULV(2,14,16)
C * * * * * CALL INTV(16,17)
C * * * * * CALL INTV(17,18)
C * * * * * CALL QVEC(18,19)
C * * * * * CALL MULV(9,17,20)
C * * * * * CALL MULV(5,15,13)
C * * * * * CALL INTV(13,14)
C * * * * * CALL SUBV(19,20,15)
C * * * * * S=LAM**2
C * * * * * CALL MULS(15,16,S)
C * * * * * S=LAM
C * * * * * CALL MULS(14,17,S)
C * * * * * CALL SUBV(16,17,18)
C * * * * * S=B(2)
C * * * * * CALL MULS(18,19,S)
BRK14600
BRK14610
BRK14620
BRK14630
BRK14640
BRK14650
BRK14660
BRK14670
BRK14680
BRK14690
BRK14700
BRK14710
BRK14720
BRK14730
BRK14740
BRK14750
BRK14760
BRK14770
BRK14780
BRK14790
BRK14800
BRK14810
BRK14820
BRK14830
BRK14840
BRK14850
BRK14860
BRK14870
BRK14880
BRK14890
BRK14900
BRK14910
BRK14920
BRK14930
BRK14940
BRK14950
BRK14960
BRK14970
BRK14980
BRK14990
BRK15000
BRK15010
BRK15020
BRK15030
BRK15040
BRK15050
BRK15060
BRK15070

```


CALL ADDV(12,19,13)	BRK15080
CALL DUPV(13,12)	BRK15090
* * * * *	BRK15100
* * * * *	BRK15110
* * * * *	BRK15120
* * * * *	BRK15130
NOW B3 PORTION.	BRK15140
* * * * *	BRK15150
* * * * *	BRK15160
CALL MULV(2,9,13)	BRK15170
CALL INTV(13,14)	BRK15180
CALL INTV(14,15)	BRK15190
CALL QVEC(15,16)	BRK15200
CALL MULV(9,14,17)	BRK15210
CALL SUBV(16,17,18)	BRK15220
S=LAM**2	BRK15230
CALL MULS(18,19,S)	BRK15240
CALL MULV(5,6,13)	BRK15250
CALL INTV(13,14)	BRK15260
S=LAM	BRK15270
CALL MULS(14,15,S)	BRK15280
CALL SUBV(19,15,20)	BRK15290
S=B(3)	BRK15300
CALL MULS(20,19,S)	BRK15310
CALL ADDV(12,19,13)	BRK15320
CALL DUPV(13,12)	BRK15330
* * * * *	BRK15340
* * * * *	BRK15350
* * * * *	BRK15360
NCW B4 PORTION.	BRK15370
* * * * *	BRK15380
* * * * *	BRK15390
* * * * *	BRK15400
CALL INTV(7,13)	BRK15410
CALL MULV(2,13,14)	BRK15420
CALL INTV(14,15)	BRK15430
CALL INTV(15,16)	BRK15440
CALL QVEC(16,17)	BRK15450
CALL MULV(9,15,18)	BRK15460
CALL ADDV(18,10,19)	BRK15470
CALL SUBV(19,17,20)	BRK15480
S=LAM**2	BRK15490
CALL MULS(20,13,S)	BRK15500
S=B(4)	BRK15510
CALL MULS(13,14,S)	BRK15520
CALL ADDV(12,14,15)	BRK15530
CALL DUPV(15,12)	BRK15540
	BRK15550

C	SUBROUTINE CIVV(N1,N2,N3) REAL*8 X(30,101),ALFA(12),G(4),S COMMON X,ALFA,G,N,KP DO 1 I=1,N 1 X(N3,I)=X(N1,I)/X(N2,I) RETURN END SUBROUTINE ADDS(N1,N2,S) REAL*8 X(30,101),ALFA(12),G(4),S COMMON X,ALFA,G,N,KP DO 1 I=1,N 1 X(N2,I)=X(N1,I)+S RETURN END SUBROUTINE SUBS(N1,N2,S) REAL*8 X(30,101),ALFA(12),G(4),S COMMON X,ALFA,G,N,KP DO 1 I=1,N 1 X(N2,I)=X(N1,I)-S RETURN END SUBROUTINE MULS(N1,N2,S) REAL*8 X(30,101),ALFA(12),G(4),S COMMON X,ALFA,G,N,KP DO 1 I=1,N 1 X(N2,I)=X(N1,I)*S RETURN END SUBROUTINE DIVS(N1,N2,S) REAL*8 X(30,101),ALFA(12),G(4),S COMMON X,ALFA,G,N,KP DO 1 I=1,N 1 X(N2,I)=X(N1,I)/S RETURN END SUBROUTINE PRIV(N1,I,J) REAL*8 X(30,101),ALFA(12),G(4),S COMMON X,ALFA,G,N,KP IF SECOND ARGUMENT EQUALS 0, GO DIRECTLY TO RETURN IF SECOND ARGUMENT EQUALS 1, PRINT THE VECTOR. IF SECOND ARGUMENT EQUALS 2, PRINT THE IDENTITY AND THE VECTOR. IF SECOND ARGUMENT EQUALS 3, PRINT THE VECTOR NUMBER AND THE VECTOR.	BRK16520 BRK16530 BRK16540 BRK16550 BRK16560 BRK16570 BRK16580 BRK16590 BRK16600 BRK16610 BRK16620 BRK16630 BRK16640 BRK16650 BRK16660 BRK16670 BRK16680 BRK16690 BRK16700 BRK16710 BRK16720 BRK16730 BRK16740 BRK16750 BRK16760 BRK16770 BRK16780 BRK16790 BRK16800 BRK16810 BRK16820 BRK16830 BRK16840 BRK16850 BRK16860 BRK16870 BRK16880 BRK16890 BRK16900 BRK16910 BRK16920 BRK16930 BRK16940 BRK16950 BRK16960 BRK16970 BRK16980 BRK16990
---	---	--


```

C IF SECOND ARGUMENT EQUALS 4, PRINT NUMBER, IDENTITY, AND VECTOR.
C IF SECOND ARGUMENT EQUALS 5, PRINT IDENTITY ONLY.
      IF(I.EQ.0) GO TO 10
      IF(I.EQ.1) GO TO 1
      IF(I.EQ.2) GO TO 2
      IF(I.EQ.3) GO TO 3
      IF(I.EQ.4) GO TO 4
      IF(I.EQ.5) GO TO 5
      DO 8 K=1,N
      8 WRITE(6,9) K,X(N1,K)
      9 FORMAT(30X,I5,1PE20.5)
      10 RETURN
      21 WRITE(6,21) J
      21 FORMAT(/,30X,'VECTOR WITH IDENTITY ',I5,' FOLLOWS:')
      31 WRITE(6,31) N1
      31 FORMAT(/,30X,'VECTOR NUMBER ',I5,' FOLLOWS:')
      41 WRITE(6,41) N1,J
      41 FORMAT(/,30X,'VECTOR X',I2,' WITH IDENTITY ',I5,' FOLLOWS:')
      51 WRITE(6,51) J
      51 FORMAT(/,30X,'VECTOR WITH IDENTITY ',I5,' HAS BEEN GENERATED.')
      GO TO 10
END
SUBROUTINE INTV(N1,N2)
  REAL*8 X(30,101),ALFA(12),G(4),S,ADD,NINO,NTNO,FIVO,THTO,EN,R
  COMMON X,ALFA,G,N,KP
  EN=N-1
  EN=1.D+0/EN
  NINO=EN*9.D+0/2.4D+1
  NTNO=EN*1.9D+1/2.4D+1
  FIVO=EN*5.D+0/2.4D+1
  THTO=EN*1.3D+1/2.4D+1
  R=EN/2.4D+1
  X(N2,1)=0.D+0
  X(N2,2)=NINO*X(N1,1)+NTNO*X(N1,2)-FIVO*X(N1,3)+X(N1,4)*R
  NM3=N-3
  DO 1 K=1,NM3
    KP1=K+1
    KP2=K+2
    KP3=K+3
    ACC=THTO*(X(N1,KP1)+X(N1,KP2))-R*(X(N1,K)+X(N1,KP3))
    1 X(N2,KP2)=X(N2,KP1)+ACC
    X(N2,N)=X(N2,N-1)+NINO*X(N1,N)+NTNO*X(N1,N-1)-FIVO*X(N1,N-2)
    1+X(N1,N-3)*R
  RETURN
END
BRK17000
BRK17010
BRK17020
BRK17030
BRK17040
BRK17050
BRK17060
BRK17070
BRK17080
BRK17090
BRK17100
BRK17110
BRK17120
BRK17130
BRK17140
BRK17150
BRK17160
BRK17170
BRK17180
BRK17190
BRK17200
BRK17210
BRK17220
BRK17230
BRK17240
BRK17250
BRK17260
BRK17270
BRK17280
BRK17290
BRK17300
BRK17310
BRK17320
BRK17330
BRK17340
BRK17350
BRK17360
BRK17370
BRK17380
BRK17390
BRK17400
BRK17410
BRK17420
BRK17430
BRK17440
BRK17450
BRK17460
BRK17470

```


BRK17480
BRK17490
BRK17500
BRK17510
BRK17520
BRK17530
BRK17540
BRK17550
BRK17560
BRK17570
BRK17580
BRK17590
BRK17600
BRK17610
BRK17620
BRK17630
BRK17640
BRK17650
BRK17660
BRK17670
BRK17680
BRK17690
BRK17700
BRK17710
BRK17720
BRK17730
BRK17740
BRK17750
BRK17760
BRK17770
BRK17780
BRK17790

```

C      SUBROUTINE DUPV(N1,N2)
      REAL*8 X(30,101),ALFA(12),G(4),S
      COMMON X,ALFA,G,N,KP
      DO 1 I=1,N
      1 X(N2,I)=X(N1,I)
      RETURN
      END
C      SUBROUTINE NORV(N1)
      REAL*8 X(30,101),ALFA(12),G(4),S
      COMMON X,ALFA,G,N,KP
      CALL INTV(N1,20)
      IF(DABS(X(20,N)).GT.1.D-10) GO TO 2
      DO 1 I=1,N
      1 X(19,I)=DABS(X(N1,I))
      CALL INTV(19,20)
      CALL DIVS(N1,18,X(20,N))
      CALL DUPV(18,N1)
      RETURN
      END
C      SUBROUTINE QVEC(I,J)
      REAL*8 X(30,101),ALFA(12),G(4),S
      COMMON X,ALFA,G,N,KP
      CALL MULV(5,I,20)
      CALL INTV(20,19)
      CALL MULV(8,I,20)
      CALL ADDV(19,20,J)
      RETURN
      END
C

```


BIBLIOGRAPHY

1. Brock, J. E., 1978: A General Procedure for Lateral Buckling of Beams, to appear as a Naval Postgraduate School Technical Report.
2. Timoshenko, S. P. and J. M. Gere, 1961: Theory of Elastic Stability, 2nd edition, McGraw-Hill Book Company.
3. Cole, J. D., 1968: Perturbation Methods in Applied Mathematics, Blaisdell Publishing Co., Watham, Mass.
4. Moler, C. B. and G. W. Stewart, 1973: "An Algorithm for Generalized Matrix Eigenvalue Problems," SIAM Journal on Numerical Analysis, Vol. 10, No. 2, April 1973, pp. 241-256.
5. Milne, W. E., 1949: Numerical Calculus, Princeton University Press.
6. Moler, C. B. and G. W. Stewart, 1971: An Algorithm for the Generalized Matrix Eigenvalue Problem, Computer Science Department, Stanford University.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
4. Professor J. E. Brock, Code 69Bc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
5. LCDR Robert E. Brown, USN 7812 Livingston Road Oxon Hill, Maryland 20021	1
6. LCDR John Michael Fahey, USN Student Mail Center Box #1709 Naval Postgraduate School Monterey, California 93940	1

Thesis

B82237

c.1

Brown

176508

Lateral buckling of
beams without warping
rigidity.

Thesis

B82237

c.1

Brown

176508

Lateral buckling of
beams without warping
rigidity.

thesB82237

Lateral buckling of beams without warpin



3 2768 002 08030 1

DUDLEY KNOX LIBRARY